



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

AF/2127
\$2000

In application of: **George et al.**

Serial No.: **09/299,952**

Filed: **April 26, 1999**

For: **Method and System for
Capturing and Storing System
Changes for Application to Multiple
Users and Systems in a Heterogeneous
Server Environment**

35525

PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

§
§
§
§
§
§
§
§
§
§

Group Art Unit: **2127**

Examiner: **Vo, Lillian**

Attorney Docket No.: **AT9-99-174**

RECEIVED
APR 28 2004
Technology Center 2100

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on April 20, 2004.

By:

Carrie Parker
Carrie Parker

TRANSMITTAL DOCUMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
ENCLOSED HEREWITH:

- Appellant's Brief (in triplicate) (37 C.F.R. 1.192); and
- Our return postcard.

A fee of \$330.00 is required for filing an Appellant's Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

Respectfully submitted,

Duke W. Yee
Duke W. Yee
Registration No. 34,285
YEE & ASSOCIATES, P.C.
P.O. Box 802333
Dallas, Texas 75380
(972) 367-2001
ATTORNEY FOR APPLICANTS

Docket No. AT9-99-174

PATENT



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Re application of: George et al.

Serial No. 09/299,952

Filed: April 26, 1999

For: Method and System for
Capturing and Storing System
Changes for Application to Multiple
Users and Systems in a Heterogeneous
Server Environment

§
§ Group Art Unit: 2127
§
§ Examiner: Vo, Lilian
§
§
§
§

RECEIVED
APR 28 2004
Technology Center 2100

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

ATTENTION: Board of Patent Appeals
and Interferences

Certificate of Mailing Under 37 C.F.R. § 1.8(a)

I hereby certify this correspondence is being deposited with the United States Postal Service as First Class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on April 20, 2004.

By: Carrie Parker
Carrie Parker

APPELLANT'S BRIEF (37 C.F.R. 1.192)

This brief is in furtherance of the Notice of Appeal, filed in this case on February 20, 2004.

The fees required under § 1.17(c), and any required petition for extension of time for filing this brief and fees therefore, are dealt with in the accompanying TRANSMITTAL OF APPEAL BRIEF.

This brief is transmitted in triplicate. (37 C.F.R. 1.192(a))

04/26/2004 SDIRETA1 00000092 090447 09299952

01 FC:140E 330.00 DA

REAL PARTIES IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines, Inc.

RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interference's that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interference's.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

Claims in the application are: 2-5, 7-13, 15-18, and 20-26.

B. STATUS OF ALL THE CLAIMS IN APPLICATION

1. Claims canceled: 1, 6, 14, 19, and 27-29.
2. Claims withdrawn from consideration but not canceled: None.
3. Claims pending: 2-5, 7-13, 15-18, and 20-26.
4. Claims allowed: None.
5. Claims rejected: 2-5, 7-13, 15-18, and 20-26.

C. CLAIMS ON APPEAL

The claims on appeal are: 2-5, 7-13, 15-18, and 20-26.

STATUS OF AMENDMENTS

No amendments were made in response to the final Office action.

SUMMARY OF INVENTION

A method and system for capturing and storing user and system changes for application to multiple users and systems in a heterogeneous server environment is provided. Page 6, lines 5-8. A data processing system is initialized for a capture of an initial state of the data processing system. Page 6, lines 8-10. The data processing system is modified. Page 6, line 10. The modified state of the data processing system is captured. Page 6, lines 10-11. The differences between the initial state and the modified state are stored as a set of configuration parameters in a depository, and the set of configuration parameters may be used to manage configurability of a data processing system within the distributed data processing system. Page 6, lines 12-18. The present invention also captures and splits the changes into between user-specific and system-specific changes so that the system changes may be applied per machine and the user changes may be applied per user. Page 15 line 29-page 16 line 4.

ISSUES

1. Whether claims 3-5, 7-13, 16-18, and 20-26 are anticipated by Piazza et al. (USPN 6026438); and
2. Whether claims 2 and 15 are unpatentable over Piazza et al. (USPN 6026438) in view of Sondur et al. (USPN 6282568).

GROUPING OF CLAIMS

The claims stand or fall together.

ARGUMENT

Applicant respectfully submits that Piazza does not teach the limitations of at least claim 7. Claim 7 is reproduced for reference:

7. A method for identifying and storing changes to a data processing system within a distributed data processing system, the method comprising the computer-implemented steps of:

- initializing the data processing system for a capture of an initial state of the data processing system;

- modifying the data processing system;

- capturing a modified state of the data processing system; and

- storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes;

- wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis;

- wherein the differences between the initial state and the modified state comprise differences between user files, system files, user registries, and system registries; and

- wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user basis.

In the final Office action dated 11/20/03, Examiner provides the reasoning behind rejection of the claims and responds to Applicant's previous arguments:

Regarding claim 7, Piazza et al. disclose a method for identifying and storing changes to a data processing system within a distributed data processing system, the method comprising the computer-implemented steps of:

- initializing the data processing system for a capture of an initial state of the data processing system (col. 3, lines 44-50, col. 6, lines 63-67);

- modifying (col. 4, lines 23-27);

- capturing the modified state (col. 6, lines 63-67, col. 7, lines 15-62, col. 8, lines 30-33);

- storing differences between initial state and the modified state (col. 7, lines 15-39), wherein the differences are separated into system-specific changes (col. 7, lines 15-62, col. 8, lines 30-32), and user-specified changes (col. 11, lines 5-65);

- wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis (col. 7, lines 15-62, col. 11, lines 5-65);

- wherein the differences between the initial state and the modified state comprise differences between user files, system files, user registries, and system registries (col. 6, lines 63-67, col. 7, lines 13-62, col. 8, lines 27-32, col. 11, lines 5-65, col. 10, lines 2-55); and

- wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user bases (col. 11, line 5-col. 12, line 36, *"This will document personal templates, address book, contact list, and other personal data. Specifically, under Profile in the home directory, file USER.USR (in NT 4.0, this file is called NTUSER.DAT) stores all USER specific registry entries which kd defined by HKEY.sub.—"*

CURRENT.sub.—USER whenever a user logs onto a workstation.”)

1. Piazza fails to teach the claimed invention because Piazza does not derive user-specific changes from the snapshot and does not separate user-specific changes from system-specific changes.

Claim 7 includes the limitations, “storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes....” To support this rejection,

In support of rejecting this claim language, Examiner points to column 7, lines 15-62 for separating differences between the initial and modified states into *system* specific changes, and to col. 11, lines 5-65 for *user*-specific changes.

However, for Piazza’s capture of data (*i.e.*, Piazza’s “snapshot”), there is no separation between system-specific changes and user-specific changes taught or suggested. For example, col. 11, lines 5-65 discuss that NT retains configuration files relevant to user preferences (col. 11, lines 28-29), but does not discuss applying user-specific changes from the initial and modified states separately from system-specific changes from the initial and modified states. That is, in Piazza, the user data does not appear to be derived from the snapshot.

Examiner also cites col. 8, lines 30-32, and col. 11, lines 50-60. The col. 8 citation is reproduced below:

The resulting system file changes are then discovered, 320, by taking a snapshot of the new files, which is stored as a Diff file, block 330.

Though this passage mentions storing system file changes by taking a snapshot, it does not appear to teach or suggest the claimed limitation of, “...wherein the differences are separated into system-specific changes and user-specific changes...” as claimed in claim 7. Likewise, col. 11, lines 50-60 state:

Specifically, TGAPINST operates on the branch server, containing the home directory, by attaching each USER’s USER.USR file to the server’s registry and then applying the

registry entries from the PCURVER file to the attached USER.USR file. TGAPINST also creates the USER's default home directory, including files, ini, etc. and sets of permission allowing the USER and only the USER access. PCURVERS file is created by removing all references to HKEY_CURRENT_USER and all references to the home directory from the previously generated workstation diff files.

This passage appears to discuss saving information related to a user, it does not teach or suggest that, as part of storing differences between initial and modified states, system-specific and user-specific changes are separated.

2. Piazza teaches away from claim 7 by teaching that all attributes in an application at any given state are aggregated into a single curver file. Piazza compares curver files from initial and modified states, but the curver files do not separate system-specific changes from user-specific changes. Hence, Piazza does not teach separating differences into system-specific and user-specific changes.

It is respectfully submitted that Piazza teaches away from the claimed limitation. For example, at col. 12, lines 10-16, Piazza discusses contrasting "Curver" files for differences:

At test 960, the system determines whether a Diff (between old, current, and new Curvers files) is needed. If yes, logic branches to block 970, and the two Curvers files are dynamically contrasted (Diff) and the output Curver file captured. If no Diff required ("No" to test 960), then the new Curver file is captured. At block 1000, the captured Curver file is applied and logic terminates at block 1010.

Hence, though Piazza does appear to teach capturing the system in different states and comparing before and after snapshots by the use of Curver files, it does not teach that system and user-specific changes are separated. Since Piazza does not teach that system- and user-specific changes are separated, Piazza fails to teach the claimed limitation of, "storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes," as claimed in at least claim 7.

3. Piazza does not teach *applying* system-specific changes on a per-system basis and user-specific changes on a per-user basis.

Further, it is respectfully submitted that Piazza does not appear to teach that system specific changes are *applied* on a per-system basis and user specific changes are applied on a per-user basis, as taught by claim 7. It actually teaches away from the present invention, by saying that the users are separated from workstations (by treating users as nodes) so that user configuration won't change under practice of their invention when system specific changes are made. Hence, the changes from Piazza's snapshot, represented in curver files, are not used to separately apply user-specific changes. Piazza therefore fails to teach the claimed limitation of, storing differences between the initial state and the modified state... wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis as claimed in at least claim 7.

Piazza address the problem of installing software on a workstation, which requires the user to log onto the workstations. For example, col. 11, lines 37-49, cited by Examiner, states:

Traditionally, USER.USR data was only entered onto the workstation while a USER was logged on and then uploaded by NT to the home directory when the USER logged off. The difficulty with this method is that it requires actual individual USERS to log on to each workstation to load software and configure USER.USR. This is unacceptable for unattended updates involving more than a handful of machines at a time.

The present invention avoids this problem by treating USERS as separate nodes (similar to workstations) which breaks the requirement that the USER is configured during a software install on a workstation...

Piazza then goes on to explain the benefits of its teachings:

Significant flexibility results from this arrangement. For example, because USER data is not created during workstation install, each workstation remains generic and available to all USERS; this allows for a mismatch between the number of USERS and workstations.

In the "Response to Arguments" section on page 2 of the Office action of 11/20/03, Examiner notes that, "the two distinct types of changes as illustrated by Piazza "inherently" indicates the separation between use-specific and system-specific changes."

Applicant respectfully disagrees. It is respectfully submitted that Examiner has misapplied the concept of "inherent" anticipation. Section 102 of Title 35 deals with novelty and loss of patent rights. An invention is said to be "anticipated" when it is squarely described or disclosed in a single reference as identified from one of the categories of 35 U.S.C. § 102, commonly referred to as "prior art". Express anticipation occurs when the invention is expressly disclosed in the prior art, patent or publication. In some cases, however, when the claimed invention is not described *in haec verba*, the "doctrine of inherency" is relied on to establish anticipation. Under the principles of inherency, a claim is anticipated if a structure in the prior art necessarily functions in accordance with the limitations of a process or method claim. *In re King*, 801 F.2d 1324, 231 U.S.P.Q. 136 (Fed. Cir. 1986). A prior art reference that discloses all of a patent's claim limitations anticipates that claim even though the reference does not expressly disclose the "inventive concept" or desirable property the patentee discovered. *Verdgaal Brothers, Inc. v. Union Oil Company of California*, 814 F.2d 628, 2 U.S.P.Q.2d 1051, (Fed. Cir. 1987). It suffices that the prior art process inherently possessed at that property. *Id.* Mere possibilities or even probabilities, however, are not enough to establish inherency. The missing claimed characteristics must be a "natural result" flowing from what is disclosed. *Continental Can Co. v. Monsanto Co.*, 948 F.2d 1264, 20 U.S.P.Q.2d 1746 (Fed. Cir. 1991). Unstated elements in a reference are inherent when they exist as a "matter of scientific fact". *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 7 U.S.P.Q.2d 1057 (Fed. Cir.), *cert. denied*, 488 U.S. 892 (1988) and *Hughes Aircraft Co. v. United States*, 8 U.S.P.Q.2d 1580 (Ct. Cl. 1988). Otherwise, the invention is not inherently anticipated.

In the present case, though Piazza appears to teach treating users (as well as workstations) as nodes, which allows workstations to remain generic to users when workstations are altered, this does not that user and system specific changes are made and applied on a per-user and a per-system basis, respectively, and it particularly fails to teach that differences between the initial and modified states (which include system- and user-specific changes) are applied separately with respect to system-specific changes and user-specific changes. No description of both user and system changes is found in Piazza; rather, Piazza only discusses how to avoid modifying user configuration when making system-specific changes (*i.e.*, by treating users as nodes, as recited above). It is reiterated that the curver files, which are used to compare before and after shots of the system, do not separate system-specific changes from user-specific changes.

Examiner also cites Piazza at col. 11 line 5-col. 12, line 36, quoting in part the following language:

This will document personal templates, address book, contact list, and other personal data. Specifically, under profile in the home directory, file USER.USR (in NT 4.0, this file is called NTUSER.DAT) stores all USER specific registry entries which is defined as HKEY.sub.--CURRENT.sub.--USER whenever a user logs onto a workstation.


[Emphasis added by Examiner.]

Examiner cites this language as teaching the claimed limitations of, “wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user basis.”

However, it is noted that the recited language from Piazza actually does not teach using “differences between user registries,” to manage configurability. The passage recited by Examiner only teaches storing personal data associated with the user. It does not appear to teach or suggest the claim limitations of applying system-specific changes on a per-system basis and applying user-specific changes on a per-user basis.

Therefore, it is respectfully submitted that the present claim 7 is distinguished from the cited reference. Further, independent claim 20 includes limitations similar to those of claim 7, was rejected under the same reasoning as was claim 7, and is believed allowable for the reasons stated above with respect to claim 7.

Therefore, all independent claims are believed distinguished from the cited reference. Likewise, because of their dependence on allowable claims, it is respectfully submitted that all dependent claims are also distinguished from the cited reference. Hence, it is respectfully submitted that the claims of the present application are in condition for allowance. Favorable reconsideration of the claims is respectfully requested.



Patrick C. R. Holmes
Reg. No. 46,380
YEE & ASSOCIATES, P.C.
PO Box 802333
Dallas, TX 75380
(972) 367-2001

APPENDIX OF CLAIMS

The text of the claims involved in the appeal are:

2. The method of 7 wherein the distributed data processing system is a heterogeneous client-server system.
3. The method of 7 wherein the data processing system is a Windows-based system.
4. The method of 7 wherein a state of the data processing system is captured by performing a snapshot of data within the data processing system.
5. The method of 4 wherein the snapshot may be configured to include or to exclude portions of data within the data processing system.
7. A method for identifying and storing changes to a data processing system within a distributed data processing system, the method comprising the computer-implemented steps of:
 - initializing the data processing system for a capture of an initial state of the data processing system;
 - modifying the data processing system;
 - capturing a modified state of the data processing system; and
 - storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes;

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis;

wherein the differences between the initial state and the modified state comprise differences between user files, system files, user registries, and system registries; and

wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user basis.

8. The method of 7 wherein the differences between system files and differences between system registries may be used to manage configurability of the application on a per-system basis.

9. The method of 7 wherein the differences between the initial state and the modified state comprise differences between .INI files.

10. The method of 9 wherein the differences between .INI files is captured line-by-line.

11. The method of 7 wherein the data processing system is modified by installing an application.

12. The method of 7 wherein the data processing system is modified by changing a registry file.

13. The method of 7 wherein the data processing system is modified by changing a .INI file.

15. The apparatus of 20 wherein the distributed data processing system is a heterogeneous client-server system.

16. The apparatus of 20 wherein the data processing system is a Windows-based system.

17. The apparatus of 20 wherein a state of the data processing system is captured by performing a snapshot of data within the data processing system.

18. The apparatus of 17 wherein the snapshot may be configured to include or to exclude portions of data within the data processing system.

20. An apparatus for identifying and storing changes to a data processing system within a distributed data processing system, the apparatus comprising:

initializing means for initializing the data processing system for a capture of an initial state of the data processing system;

modifying means for modifying the data processing system;

capturing means for capturing a modified state of the data processing system; and

storing means for storing differences between the initial state and the modified state as a set of configuration parameters in a depository, wherein the differences are separated into system-specific changes and user-specific changes;

wherein the system specific changes are applied on a per-system basis and the user-specific changes are applied on a per-user basis;

wherein the differences between the initial state and the modified state comprise differences between user files, system files, user registries, and system registries; and wherein the differences between user files and differences between user registries may be used to manage configurability of the application on a per-user basis.

21. The apparatus of 20 wherein the differences between system files and differences between system registries may be used to manage configurability of the application on a per-system basis.

22. The apparatus of 20 wherein the differences between the initial state and the modified state comprise differences between .INI files.

23. The apparatus of 22 wherein the differences between .INI files is captured line-by-line.

24. The apparatus of 20 wherein the data processing system is modified by installing an application.

25. The apparatus of 20 wherein the data processing system is modified by changing a registry file.

26. The apparatus of 20 wherein the data processing system is modified by changing a .INI file.